Network Service Sharing Infrastructure: Service Authentication and Authorization Revocation

DAVID LAI & ZHONGWEI ZHANG
Department of Mathematics and Computing
University of Southern Queensland
Toowoomba, Queensland, 4350
AUSTRALIA
{lai,zhongwei}@usq.edu.au

Abstract: - When a service server receives a request, the server will establish the identity and authorization of the user based on the information stored in authentication information repository (IDAR) before service is provided. The IDAR will determine who can have access to which service. A legitimate user must have her/his identity and authorization registered in the IDAR in advance. Users who registered in IDAR of another server or network cannot access services in another server or network. This prevents effective and efficient sharing of services.
In this paper, we develop a Network Service Sharing Infrastructure (NSSI) by which many networks are linked together for service sharing. This *ad hoc* network system can provide a wider range of services to users than any individual network. Within the *ad hoc* network system, individual networks authenticate and grant authorizations independent of each other by using their own IDAR. NSSI enables authentication and authorization results to be relayed to other linked networks to access a shared services while individual networks still maintain their own authentication scheme or authentication requirements. This makes joining and leaving NSSI simple and involves minimum administrative overhead.

Key-Words: - sharing, service sharing, service authentication, service path, authentication propagation, authentication token.

1 Introduction

It is neither efficient nor secure to let one network to offer too many services. A network can expand its range of services with services provided by other networks. When a service is shared with another network, the server has to handle authentication, authorization and its revocation of an user who has registered with IDAR other than the one used by the server. It means that the server has to find the right IDAR and resolve how to retrieve the information in correct format.

The situation becomes more complicated as autonomous networks are linked together for sharing services. Servers may not be able to access IDARs in another autonomous network. To obtain a service, user may have to register with many networks, and log on to different networks for individual service.

As servers may be downed for maintenance and networks may join in or leave the service sharing system dynamically, it is important for users to have a complete and current list of local and shared services available. Further more, if the authorization of a user is changed or revoked at some stage during a service session, the server providing the service should know the change or revocation.

Various methods such as the use of X.509 certificates [11], trust recommendations [5] [6], trust establishment [1] [2] [7] [8] [15] and Kerberos [9] were proposed as possible solutions to the problems. The major concerns about these solutions are the freshness of certificates, establishing a trusted common third parted and static configuration of the networks in Kerberos.

It is good if there is an infrastructure in which autonomous networks can link together for sharing services with minimum initial set up overheads and using local authentication for both local and shared services. Under this infrastructure, users can inquire about local services and shared services from an agent in the local network. And servers will be notified of any change in user authorizations for login sessions.

In this paper, we further develop Network Service Sharing Infrastructure (NSSI) which uses will enable autonomous networks to use local authentication for shared services. Users can query information about shared services available within this infrastructure.

The rest of the paper is organized as follows. In section 2, the major issues of service authentication and related works are identified. In section 3, we outline NSSI along with Service Network Graph and Distributed Network Service Authentication protocol. In section 4, we will discuss how changes in authorization

and revocation are propagated to servers concerned under the NSSI. We will conclude the paper in section 5 with a discussion on future work.

2 Problems of Service Authentication and Related Works

Service authentication is a process of establishing the identity of a user who requests service on a network. In the simplest case where there is only one server with its own IDAR, service authentication is just a log-in session to the server. In such a login session, a user needs to provide a set of authentication information to the server and gets the appropriate authorization. What a user has to collect and maintain is a single set of authentication information.

Among the servers within a network, some will have their own IDAR while others will share an IDAR as a group. The format and content of each record stored in an IDAR may vary from one IDAR to another even for the same user.

The number of different authentication records for an user in all those IDARs are the number of authentication information sets the user has to maintain. Even in the cases where all servers share the same IDAR, or all IDARs have authentication records of the same format and content, the user still has to log in to each server independently to access services from individual servers.

When autonomous network link together for service sharing, they form a graph of networks. We will refer to such a graph of network Service Network Graph (SNG). Each node in a SNG represents an autonomous network participating in the sharing of services.

When a SNG is formed, each autonomous network will have its own IDAR and authentication information are not shared. Network administrators face the problem of authenticating users from other networks which have various authentication schemes and authentication information sets. It is obvious that enforcing a common authentication scheme is not feasible and involves substantial administrative overheads. For instance, when a network using an authentication scheme different from the common authentication scheme links to a SNG, it has to switch to the common authentication scheme. All users of the network have to collect and use a new set of authentication information. When the network detaches from the SNG, it has to choose between reverting back to the original authentication scheme or stay with the common authentication scheme used by the SNG. Obviously, if the initial adoption of the original authentication scheme by the network has its own reasons, and those reasons are still valid, the network is going to revert to the original authentication scheme. The administrative overhead and possible confusion and frustration among the users are not to be undermined.

If individual networks do not share their authentication data, users must register themselves with each server or network they wish to access. Maintaining a global set of authentication data is deemed to fail as some networks may be reluctant to disclose the authentication data for security reasons. It is even worse that some networks may link to the graph or detached from the graph at any time. As a result, setting up a global authentication set is practically infeasible. A typical example is the X.500 [10] plan which has never succeeded in producing a global database of named entities.

Kerberos [9] represent a solution in which users authenticate with a central authentication server and the authentication status can be relayed to the required servers. With one set of authentication information and one log-in, users will be able to access services available from all servers within the same network. Service sharing is achieved by static links between individual realms. It does not handle dynamic linking of networks efficiently.

Another solution to this problem is ISO X.509 [11] recommendation which was published in 1993. Authentication in X.509 is based on the secrecy of the private key and the binding of the public key to a user name by a Certificate Authority (CA). The center of this authentication mechanism is the trust for the Certificate Authority.

Note that an administrator of an autonomous network may decide to set up a CA for the network or empower a third party to run the CA. However, when many autonomous networks form a SNG, they must agree on a common CA to issue all certificates or on CA certificate chaining. We envisage that the workload increases with the number of users involved.

Another approach is to establish a trust [1] [2] [7] [8]. Trust is the result of an assessment of an entity relative to a domain of action [4] by an observer. When an observer is authorized by a network administrator to give trust recommendations [5] [6], the observer becomes a trust agent. The trust is represented by a token and each trust token is signed by the trust agent.

It is reasonable for each autonomous network to have its own set of independent trust agents. A user will be asked to provide trust tokens from a few trust agents. By using the aggregated result [3] of the trust tokens,

the server can determine the authentication and authorization status of the user for the requested service.

This works fine for individual networks. However, for SNG, each autonomous network will have its own set of trust agents. Either all the networks adopt the same common set of trust agents or the user has to collect trust tokens from different sets of trust agents for services outsourced by different networks.

It is desirable to establish an infrastructure for service sharing which allow autonomous networks link and detach from a SNG with minimum administrative overhead while retaining their own autonomy, independence and integrity. At the same time, forming a SNG should involve no extra input from users. In order words, users should not be involved in the service sharing process.

Our research aims at devising an authentication protocol and developing a service sharing infrastructure for a SNG which allows: (1) Each node can have different authentication scheme of its own. (2) Each node maintains its own IDAR. (3) service authentication can be performed locally at each node, but the authentication status will be relayed automatically to other nodes in the SNG. (4) A current list of local and shared services is available to all users. (5) Revocation of authorization is propagated to servers concerned.

3 Network Service Sharing Infrastructure

We outline the Network Service Sharing Infrastructure (NSSI) in *ad hoc* SNG using Distributed Networks Service Authentication Protocol (DNSA) in this section. The concept of SNG is presented first and then followed by the DNSA protocol.

3.1 Service Network Graph

Service Sharing infrastructure is based on SNG and service paths. An autonomous network is assumed to consist of the following entities - an Authentication Server (AS) which authenticates local users, a Server (S) which provides services, a Service Locating Server (SLS) which stores information about local services and shared services and a number of local users (U). We also assume that an encrypted channel authenticates statements transmitted via the channel [14]. All communications among autonomous networks and between hosts within the same network are assumed to be encrypted using the symmetric encryption and symmetric encryption and decryption keys. For example, Server Key (K^s) is the encryption and decryption key shared between AS and S while Session Key (K^u) is the encryption and decryption key shared between S and U and generated for each nondiscriminatory log-in session. Note that the network which provides the actual shared service is the *target network*.

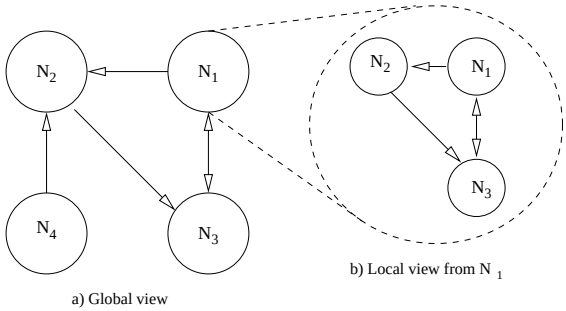

Figure 1: A Service Graph

Next we define that N_1 is attached to N_2 when N_2 delegates its authentication authority to N_1. In which case AS_2 generates and shares ATK^2 with AS^1. N_2 is the delegator network and N_1 is the delegatee network and N_2 provides services to N_1 as outsourced services of N_1. This is a one-way relationship and is represented by a single arrow as shown in Figure 1.

We also define N_1 and N_2 are mutually linked when N_1 is attached to N_2 and N_2 is also attached to N_1. Mutual linking is therefore a dual-way relationship.

A *Service Network Graph* is a set of networks attached to or mutually linked with each other. We will use a single arrow to represent an attachment and a double-end arrow to represent a mutual-link as depicted in Figure 1a. A local view is the SNG of all reachable nodes as seen from a particular node. A local view from N_1 is shown in Figure 1b.

With the Network Service Sharing Infrastructure in place, we can now proceed to look at the Distributed Networks Service Authentication Protocol.

3.2 Distributed Networks Service Authentication Protocol

The Distributed Networks Service Authentication Protocol has two distinct operation modes. One is the *Network Participation mode* (NP mode) in which a network links to another network in a SNG. Another is the *User Service mode* (US mode) in which a user access a local or shared service. We will discuss them in the following sections.

3.2.1 Protocol in NP Mode

Let us assume N_1 and N_2 are two separate autonomous networks as shown in Figure 2. Upon receiving the re-

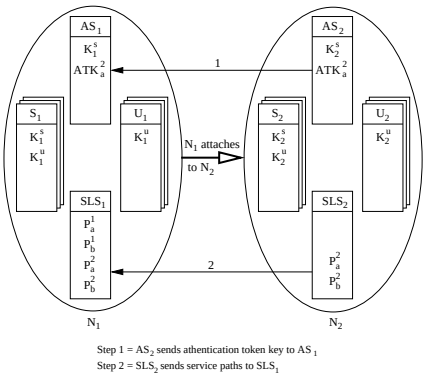

Figure 2: Attaching one network to another network

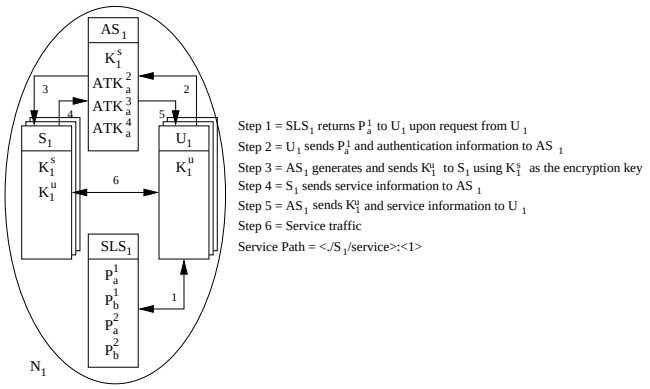

Figure 3: User requesting a local service

quest from N_1 to directly attach to N_2, AS_2 generates and sends ATK_a^2 to AS_1. SLS_2 also sends information about services available for sharing to SLS_1 in the form of a Service Path (P). A Service Path is a service locater similar to an URL and is represented by a string of network path and costing metrics. An example is

< [NetworkPath/]TargetNetwork/Server/Service>:<CostMetrics>

where [**Network Path/**] is optional. Services available for sharing includes local services in N_2 and those outsourced by N_2. Both AS_1 and SLS_1 will acknowledge the receipt of information to AS_2 and SLS_2 respectively. From the information of service paths received, SLS_1 can work out its own set of service paths. Note that N_2 can also request to attach to N_1 and form a mutual link with N_1.

We will explain how to handle a service request in section 3.2.2.

3.2.2 Protocol in US Mode

When a local user U_1 in N_1 requests a service, it will first query SLS_1 whether it is available or not. SLS_1 then returns a message containing a valid service path (P_a^1 or P_a^2 for example) plus its cost metrics or "Service not available" to U_1. If U_1 is comfortable with the cost metrics, the user will authenticate itself to AS_1 and pass along the service path and cost metrics which AS_1 will use to determine the path to reach the target server.

If the authentication is successful, AS_1 will generate a session key K_1^u. If the service is available on a local server S_1 as indicated by the Service Path, AS_1 will encrypt K_1^u using encryption key K_1^s of server S_1 and sent it along with user authorization information to S_1. In both cases, AS_1 will keep a record the user request and the service path. S_1 acknowledges the session key K_1^u and returns AS_1 all service information for the request. AS_1 relays the service information and K_1^u to U_1 as shown in Figure 3.

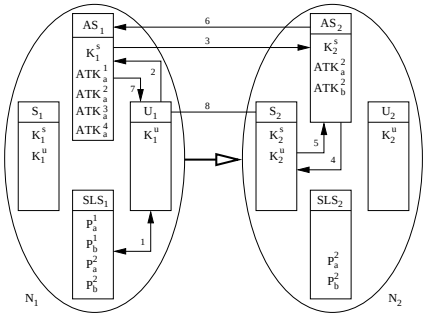

Figure 4: User requesting a shared service

If the service is available in N_2 instead of in N_1 as shown in Figure 4, AS_1 retrieves the authentication token key of AS_2, ATK_a^2 and uses it to encrypt the session key K_1^u instead of using a server encryption key K_1^s. The encrypted session key K_1^u together with the service path and user authorization information forms an *authentication token*. On receiving the authentication token from AS_1, the authentication server AS_2 in N_2 extracts the K_1^u from the authentication token. The service path embedded in the authentication token indicates that the service is offered by S_2. So AS_2 encrypts the authentication token with K_2^s and sends it to S_2 as explained before. In case when N_1 is not directly attached to N_2 as shown in Figure 5 the service path would indicate target network is only reachable via N_4. In this case K_1^u is passed on from AS_1 to AS_2 via authentication server AS_4 in N_4. AS_1 will encrypt K_1^u with ATK_a^4 while AS_4 will encrypt K_1^u with ATK_a^2. Service information returned from S_2 will follow a similar path but in the reverse order.

To tear down a service session gracefully, a server

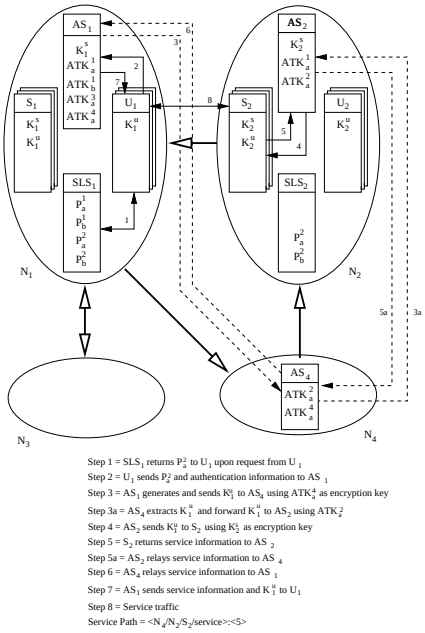

Figure 5: User requesting a shared service

will return an *End of Session* message to the AS of the network from which the service request originates. The AS will then delete the service path and user request record from its database.

3.3 Service List

When network N_2 delegates its authentication to another network N_1, SLS_2 sends all its service paths to AS_1 encrypted with ATK_2. AS_1 decrypts and forward all the service paths to SLS_1. From the service paths received, SLS_1 can work out a local view of the SNG. This local view will be used to form a set of service paths with minimal cost. Service paths from SLS_2 such as $< N_3/S_3/SQL >:< 2 >$ are optimized using the local view of SNG. Assuming a flat network cost of 2, the service path is optimized to $< N_3/S_3/SQL >:< 2 >$ instead of $< N_2/N_3/S_3/SQL >:< 4 >$. This can be derive from the local view of the SNG shown in Figure 1b. As authentication status can be relayed to N_3 directly from N_1, there is no need to go through N_2 in order to reach N_3.

Together with the service paths for local service, these service paths formed a complete service list available to users in N_1.

When a server is down for maintenance, it will send a *Server Down* message to SLSs of all attached network. The SLSs will then then update their service list by marking the corresponding services as not available. These services can be made available again when SLSs receive *Server Up* message from the server concerned.

Similarly, when an AS is down for maintenance, it will send a N_x *Down* message to SLSs of all attached network. The SLSs will then then update their service list by marking all service paths that uses N_x as not available. These services can be made available again when SLSs receive N_x *Up* message from the AS concerned.

4 Changes and Revocations in Authorization

Changes in authorization may change a user's right for services. A total revocation of authorization for a user is equivalent to unregistering the user and deny the user from any service the network may offer. In this section, we will see how changes in user authorization are propagated to servers concerned.

Let us assume that the authorization of user U_r registered with network N_1 is revoked and AS_1 is the only agent that has full access to the authentication and authorization information in the IDAR used. Any changes in authorization for U_r must involve AS_1 since it is the only agent which can alter the content of the IDAR.

When AS_1 is alert of any authorization changes for U_r, it will make changes in the IDAR. AS_1 will then check the user request records. If U_r is engaged in a service session, AS_1 will push the authorization changes to the server concerned as shown in the service path using a *revocation token*. The content of a revocation token is similar to that of an authentication token. Only the user authorization is replaced with the changed authorization or with "authorization revoked" to indicate a total revocation of authorization. The mechanism is exactly the same as the transfer of authentication token for a service session as shown in Figure 6. On receiving such an authorization change message, the server will determine the action according to the change and predefined access control policy.

5 Conclusion

A user interested in a service available in a distributed network environment has to establish a trust relationship with a local network first. Each time when a service is requested, the network has to authenticate the user before granting the user access to any local or outsourced service. A service authentication protocol which relays authentication status from local network to target network is required for shared services. In this paper, we extended Network Service Sharing Infrastructure based on Service Network Graphs, Service

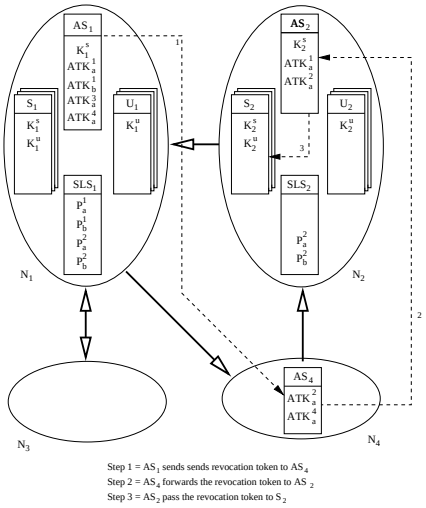

Figure 6: Changes in user authorization is pushed to server

Paths and incorporated the Distributed Network Service Authentication protocol. With NSSI, a user register with an autonomous network within an SNG can log on by using the authentication server of the network and access the shared services of other networks within the SNG. A current service list with optimized network cost is available to all users. Revocation of authorization is pushed from the AS which initiated the revocation process to the server in the service path of the user is engaged in service sharing.

We shall focus on service path optimization and security issues in the future.

References

[1] T. Beth and M. Borcherding and B. Klein, Valuation of Trust in Open Networks. *Proceedings of the Conference on Computer Security 1994*, 1994.

[2] M. Reiter and S. Stubblebine, Authentication Metric Analysis and Design. *ACM Transactions on Information and System Security*, Vol. 2, No.2, 1999.

[3] A. Abdul-Rahman and S. Halles, A Distributed Trust Model. *Proceedings of New Security Paradigms Workshops*, 1997.

[4] D. Denning, A new paradigm for trusted systems. *Proceedings of 1992-1993 ACM SIGSAC New Security Paradigms Workshop*, 1993.

[5] M. Montaner, B. Lopez and J. L. Rosa, Developing Trust in Recommender Agents. *Proceedings of the first international joint conference on Autonomous agents and multi-agent systems*, 2002

[6] S. Robles, J. Borrell, J. Bigham, L. Tokarchuk and L. Cuthbert, Design of a Trust Model for a Secure Multi-Agent Marketplace. *Proceedings of the fifth international conference on Autonomous agents*, 2001

[7] A. Abdul-Rahman and S. Hailes, Using Recommendations for Managing Trust in Distributed Systems. *Proceedings of IEEE Malaysia International Conference on Communication '97 (MICC'97), Kuala Lumpur, Malaysia*, 1997

[8] A. Abdul-Rahman and S. Hailes, Supporting Trust in Virtual Communities. *Hawaii Int. Conference on System Sciences 33 , Maui, Hawaii*, January 2000.

[9] The Kerberos Network Authentication Service (V5). *Internet Engineering Task Force (IETF) and the Internet Engineering Steering Group (IESG) Porpoised Standard, RFC1510.*

[10] X.500 (02/01). *International Telecommunication Union ITU-T Recommendations X series.* http://www.itu.int/rec/recommendation.asp

[11] X.509 (03/00). *International Telecommunication Union ITU-T Recommendations X series.* http://www.itu.int/rec/recommendation.asp

[12] M. Naor and K. Nissim, Certificate Revocation and Certificate Update. *Proceedings 7th USENIX Security Symposium (San Antonio, Texas)*, Jan 1998.

[13] S. G. Stubblebine, Recent-secure authentication: Enforcing revocation in distributed systems. *IEEE Computer Society Symposium on Security and Privacy, Oakland, California*, May 1995.

[14] B. Lampson, M. Abadi, M. Burrows and E. Wobber, Authentication in Distributed Systems: Theory and Practice. *ACM Transactions on Computer Systems*, vol. 10, no. 4, 1992.

[15] R. Au, M. Looi and P. Ashley, Automated cross-organisational trust establishment on extranets, *Proceedings of the workshop on Information technology for virtual enterprises*, 2001, page 3 - 11.

Acknowledgment: Special thanks to our colleague Walter Spunde for his constructive suggestions.